

**International
Standard**

ELF 5001

Annotated EXPRESS

**First edition
2024-05**



COPYRIGHT PROTECTED DOCUMENT

© ELF 2024

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword	iv
Introduction	v
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Basic syntax principles	2
4.1 General	2
5 Types of remarks	2
5.1 Basic objects	2
5.2 STEPmod remarks	3
6 Content syntax	3
6.1 General	3
6.2 EXPRESS object cross-references	3
7 Usage	4
7.1 Application on schemas	4
7.2 Application on entities	4
7.3 Application on other EXPRESS objects	5
Bibliography	6

Foreword

The EXPRESS Language Foundation (“ELF”) is a registered public charity in the US that facilitates the education, standardization, research, promotion, definition, and usage of information modelling and programming languages, with a focus on the EXPRESS language family.

ELF works with international partners and experts across the globe, reflecting the international nature of its mission. More information about ELF is available on the official website (<https://www.expresslang.org>).

The procedures used to develop this document and those intended for its further maintenance are described in the ELF Directives.

In particular, the different approval criteria needed for the different types of ELF documents should be noted. This document was drafted in accordance with the editorial rules of the ELF Directives.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ELF shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be provided in the Introduction.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

This document was prepared by Technical Committee *EXPRESS*.

Introduction

Annotated EXPRESS is a set of rules that build on top of the EXPRESS remark tag syntax for documentation and description of EXPRESS code. It can be considered in the same category of Doxygen for C, JavaDoc for Java, RubyDoc for Ruby.

EXPRESS is a data modelling language offered in a code-like syntax.

As in typical source code, you can insert comments that do not get parsed or interpreted by the compiler. For example, in C, block comments are realized with `/* ... */`.

In EXPRESS, the remark tag syntax is `(* ... *)`, indicating that anything in the ... is considered documentation.

Annotated EXPRESS accepts Metanorma AsciiDoc as content.

Annotated EXPRESS

1 Scope

This document specifies Annotated EXPRESS, a set of rules that build on top of the EXPRESS remark tag syntax for documentation and description of EXPRESS code.

Annotated EXPRESS provides standardized methods for documenting EXPRESS schemas, entities, properties, functions, and rules using a structured comment syntax.

This document:

- defines the structure and syntax of Annotated EXPRESS;
- specifies rules for different types of remarks that can be used in EXPRESS code;
- outlines methods for cross-referencing EXPRESS objects; and
- provides examples of proper Annotated EXPRESS usage.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 10303-11, *Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual*

AsciiDoc, AsciiDoc markup language — <https://asciidoc.org>

Metanorma, Metanorma framework for standards documents — <https://www.metanorma.org>

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org>

For the purposes of this document, the following terms and definitions apply.

3.1

EXPRESS

data modeling language defined in ISO 10303-11 that provides a formal way to specify data structures and constraints

[SOURCE: ISO 10303-11]

3.2

remark tag

syntax in EXPRESS using (* ... *) notation to indicate documentation content

[SOURCE: ISO 10303-11]

3.3

named remark tag

remark tag that includes a specific name to reference an EXPRESS object

3.4

tag reference

string identifier that uniquely identifies an EXPRESS object in the form of `schema_name.entity_name.property_name`

4 Basic syntax principles

4.1 General

Annotated EXPRESS uses the “named remark tag” syntax described in the EXPRESS language manual.

```
(* "NAME"
...
*)
```

Figure 1

Where:

— `NAME` is a “tag” that references an EXPRESS object, such as a schema, entity, property, function, or rule.

A tag looks like `schema_name.entity_name.property_name`.

EXAMPLE Given this schema `action_schema`:

```
SCHEMA action_schema;
  ENTITY action;
    name : label;
    description : OPTIONAL text;
    chosen_method : action_method;
  DERIVE
    id : identifier := get_id_value(SELF);
  WHERE
    WR1: SIZEOF(USEDIN(SELF, 'BASIC_ATTRIBUTE_SCHEMA.' + 'ID_ATTRIBUTE.IDENTIFIED_
ITEM')) <= 1;
  END_ENTITY;
END_SCHEMA;
```

The tag references would be:

- to the schema: `"action_schema"`
- to the entity `action`: `"action_schema.action"`
- to the property `name`: `"action_schema.action.name"`
- to the derived property `id`: `"action_schema.action.id"`
- to the where rule `WR1`: `"action_schema.action.wr:WR1"`

5 Types of remarks

5.1 Basic objects

In Annotated EXPRESS, there are a few types of named remarks:

- `(* "{TAG}" ... *)` refers to the description of EXPRESS object `{TAG}`

- (`"{TAG}.__note" ... *`) refers to a note that applies to EXPRESS object `{TAG}`
- (`"{TAG}.__example" ... *`) refers to a usage example that applies to the EXPRESS object `{TAG}`
- (`"{TAG}.__figure" ... *`) refers to a figure that applies to the EXPRESS object `{TAG}`, which can be re-used within its notes or examples.
- (`"{TAG}.__table" ... *`) refers to a table that applies to the EXPRESS object `{TAG}`, which can be re-used within its notes or examples.

These different types of remarks can be in multiple instances, for example, multiple (`"{TAG}" ... *`) remarks can be made in an EXPRESS file to describe the particular object.

These named remarks are not bound to any particular location in an EXPRESS file. For example, they can be made immediately after the declaration of the EXPRESS object, or collected at the top or bottom of the EXPRESS file.

5.2 STEPmod remarks

In STEPmod, which is the STEP modules library, two additional types of named remarks are used.

- (`"{SCHEMA_TAG}.__fund_cons" ... *`) describes the “Fundamental concepts and assumptions” of a schema. It is a ISO 10303 convention to describe the concepts and assumptions of the schema. The tag must reference an EXPRESS schema.
- (`"{SCHEMA_TAG}.__expressg" ... *`) provides EXPRESS-G diagrams relevant to the EXPRESS schema. These diagrams can describe an architecture view that involves the named schema. The tag must reference an EXPRESS schema.

6 Content syntax

6.1 General

Annotated EXPRESS (in its current form) accepts Metanorma AsciiDoc. The syntax of Metanorma AsciiDoc is described at <https://www.metanorma.org>.

There are several extensions to Metanorma AsciiDoc for the documentation of EXPRESS.

6.2 EXPRESS object cross-references

In Annotated EXPRESS remark content, it is often necessary to cross-reference other EXPRESS objects.

In the generated EXPRESS documentation, these references become links to the definition of the target EXPRESS objects.

The syntax is:

```
<<express:{TAG}>>
```

Figure 2

or

```
<<express:{TAG},{RENDER TEXT}>>
```

Figure 3

Where:

- `{TAG}` is the EXPRESS named tag, e.g. `action_schema.action_method`
- `{RENDER TEXT}` is the desired rendering text, if different from the named tag, e.g. `action methods`

7 Usage

7.1 Application on schemas

Let's take an example `action_schema.exp`:

```
(*"action_schema"
The subject of the *action_schema* is the description of actions, the reasons
for these actions, and the status of these actions.
*)

(*"action_schema.__fund_cons"
Action information can be attached to any aspect of product data.
*)

(*"action_schema.__example"
Reasons for action include evolving user requirements, manufacturing problems
and difficulties that arise when a product is in use.
*)

(*"action_schema.__expressg"
[.svgmap]
====
image::action_schemaexpg1.svg[]

* <<express:basic_attribute_schema>>; 1
* <<express:action_schema>>; 2
* <<express:support_resource_schema>>; 3
====
*)
```

Figure 4

- Content in `(*"action_schema" ... *)` provides a basic description (and purpose) of the schema.
- Content in `(*"action_schema.__fund_cons" ... *)` describes the concepts and assumptions in creating this schema.
- Content in `(*"action_schema.__example" ... *)` describes an example on how the schema can be used.
- Content in `(*"action_schema.__expressg" ... *)` provides a graphical diagram (in Metanorma AsciiDoc syntax with an `svg` here) relevant to the understanding of the schema.

7.2 Application on entities

Entities inside the schema are accessed using the `{schema}.{entity}` syntax (potentially multiple dots).

For example, the `action_schema.supported_item` entity is documented like this:

```
(*"action_schema.supported_item"
The *supported_item* allows for the designation of an
<<express:action_schema.action_directive,action_directive>>, an
<<express:action_schema.action,action>>, or an
<<express:action_schema.action_method,action_method>>.
*)

(*"action_schema.supported_item.__note"
This specifies the use of an
<<express:action_schema.action_resource,action_resource>>.
*)
```

Figure 5

Notice that within the named remark `action_schema.supported_item.__note`, there is an [express/...](#) link which references another EXPRESS object `action_schema.action_resource`.

7.3 Application on other EXPRESS objects

Annotations can be made to any EXPRESS objects that are referencable, including:

- ENTITY
 - properties
 - DERIVE properties
 - WHERE rules
 - IP: Informal proposition rules
- TYPE
- FUNCTION
 - LOCAL variables

Bibliography

- [1] ISO 10303-41, *Industrial automation systems and integration — Product data representation and exchange — Part 41: Integrated generic resource: Fundamentals of product description and support*
- [2] Doxygen, Documentation generator for C and other languages — link:[https://www.doxygen.nl++\[\]](https://www.doxygen.nl++[])
- [3] EXPRESS-G, *ISO 10303-11: EXPRESS-G graphical notation for EXPRESS*
- [4] JavaDoc, Java Documentation tool — <https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html>
- [5] STEPmod, STEP Modules Library — <https://www.stepmod.org>

